said program converting unit generates a machine language instruction from a source program for a processor out of an embedded-type custom processor series which has an address width N in accordance with a necessary program size, and

said processor comprises:

memory means for storing a program, the memory means having a minimum storage capacity of 2^N bytes to store the program and having N address lines, the program including an N-bit data arithmetic operation instruction and other instructions operating on both N-bit and M-bit data, N being greater than M; and

a processor core having an address bus of N bits which is equal in size to the number of address lines of the memory means, the processor core being selected from a plurality of processor cores,

wherein the processor core includes:

a program counter for holding an N-bit instruction address to output an instruction at the N-bit address to the memory means, the program counter having an N-bit address length which is equal in size to the number of address lines of the memory means;

fetching means for fetching an instruction from the memory means using an N-bit instruction address from said program counter; and

executing means for executing all N-bit arithmetic operation instructions and for executing other instructions except for arithmetic operation instructions at one of N-bit length and M-bit length, the executing means having N-bit length,

whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data bit-width, said data bit-width being M, and

said program converting unit comprises:

parameter holding means for holding a data width M and a pointer width N, said data width M representing the number of bits of data used in the source program, said pointer width N representing the number of bits of an address, said N and M being input by a user in accordance with program size; and

generating means for generating an instruction based on the source program to set the data width M as valid when a variable used in a machine language instruction to be generated is a variable showing data, and for generating an instruction to set the address width N as valid when a variable used in a machine language instruction to be generated is a variable representing an address,

wherein the program converting unit generates a unique set of machine language instructions from the source program for each N specified by the user.

50 49
~~59~~. The computer system of Claim ~~58~~, wherein the processor further comprises:

an address register group including a plurality of N-bit address registers;

a data register group including a plurality of N-bit data registers,

wherein said executing means executes the N-bit and M-bit data operation instructions using the address registers, while executing the M-bit data operation instruction using data registers.

51 50
~~60~~. The computer system of Claim ~~59~~, wherein:

said N is 24 and said M is 16; and

said processor is installed in a 1-chip microcomputer, whereby said 1-chip microcomputer becomes suitable for running a program that utilizes a memory over 64 Kbytes for an operation with 16-bit data.

52 51
~~61~~. The computer system of Claim ~~60~~, wherein the processor further comprises:

compensating means for extending an effective bit-width of the data in one of the address registers and the data register to 24 bits,

wherein said compensating means operates in accordance with a compensate instruction entered after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

53 52

62. The computer system of Claim 61, wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register; and

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

56 49

63. The computer system of Claim 58, wherein the processor further comprises:

an address register group including a plurality of N-bit address registers; and

a data register group including a plurality of M-bit data registers,

wherein said executing means executes one of an N-bit data operation instruction and an M-bit data operation instruction using the address registers, while executing the M-bit data operation instruction using the data registers.

57 54

64. The computer system of Claim 63, wherein:

said N is 24 and said M is 16; and

said processor is installed in a 1-chip microcomputer, whereby said 1-chip microcomputer becomes suitable for running a program that utilizes a memory over 64 Kbytes for an operation with 16-bit data.

58 57

65. The computer system of Claim 64, wherein the processor further comprises:

compensating means for extending an effective bit-width of the data in one of the address registers and the data register to 24 bits,

wherein said compensating means operates in accordance with a compensate instruction entered after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

59 58

66. The computer system of Claim 65, wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register;

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

54 50

67. The computer system of Claim 59, wherein the processor further comprises:

compensating means for extending an effective bit-width of the data in one of the address registers and the data register to N bits,

wherein said compensating means operates in accordance with a compensate instruction entered after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

55 54

68. The computer system of Claim 67, wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register; and

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

60 49

69. The computer system of Claim 58 wherein the pointer width N and the data width M are input by a user during an execution of the program converting unit.

61

70. A computer system comprising a central processing unit and a software program compiler, wherein

the central processing unit is one of a series of processing units, each processing unit having a different address length N, N being longer than a data width M, the address length of the processing unit selected based on a size of a source program, the processing unit comprising:

memory means for storing a program, the memory means having a minimum storage capacity of 2^N bytes to store the program and having N address lines, the program including an N-bit data arithmetic operation instruction and other instructions operating on both N-bit and M-bit data, N being greater than M; and

a processor core having an address bus of N bits which is equal in size to the number of address lines of the memory means, the processor core being selected from a plurality of processor cores,

wherein the processor core includes:

a program counter for holding an N-bit instruction address to output an instruction at the N-bit address to the memory means, the program counter having an N-bit address length which is equal in size to the number of address lines of the memory means;

fetching means for fetching an instruction from the memory means using an N-bit instruction address from said program counter; and

executing means for executing all N-bit arithmetic operation instructions and for executing other instructions except for arithmetic operation instructions at one of N-bit length and M-bit length, the executing means having N-bit length,

compensating means for extending an effective bit-width of the data in one of the address registers and the data register to N bits, wherein the compensating means compensates as directed by a compensate instruction which is entered after a machine language arithmetic instruction which may cause an overflow;

whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data bit-width, said data bit-width being M, and

the software compiler comprises:

parameter holding means for holding a data width M and a pointer width N, the data width M representing the number of bits of data used in the source program, the pointer width N representing the number of bits of an address, N and M being inputs to the compiler input by a user during an execution of the compiler, N and M selected by the user based on the size of the source program; and

generating means for generating an instruction based on the source program to set the data width M as valid when a variable used in a machine language instruction to be generated is a variable showing data, and for generating an instruction to set the address width N as valid when a variable used in a machine language instruction to be generated is a variable representing an address,

wherein the program converting unit generates a unique set of machine language instructions from the source program for each N specified by the user.

150